ESP32 Hardware Serial2 Example

[December 31, 2018](https://circuits4you.com/2018/12/31/esp32-hardware-serial2-example/)[ESP32](https://circuits4you.com/category/esp32/)[ESP32](https://circuits4you.com/tag/esp32/), [Serial2](https://circuits4you.com/tag/serial2/)

There are three serial ports on the **ESP32** known as **U0UXD, U1UXD and U2UXD** **all work at 3.3V TTL Level**. There are three **hardware supported serial interfaces** on the ESP32 known as UART0, UART1 and UART2. Like all peripherals, the pins for the UARTs can be logically mapped to any of the available pins on the ESP32. However, the UARTs can also have direct access which marginally improves performance. The pin mapping table for this hardware assistance is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UART** | **RX IO** | **TX IO** | **CTS** | **RTS** |
| UART0 | GPIO3 | GPIO1 | N/A | N/A |
| UART1 | GPIO9 | GPIO10 | GPIO6 | GPIO11 |
| UART2 | GPIO16 | GPIO17 | GPIO8 | GPIO7 |

Having said that, the UART drivers that I recommend to use don’t have this level of optimization built into them and as a result, you are pretty much free to use any pins you choose.

**Introduction to UART**

UART stands for Universal Asynchronous Receiver/Transmitter. It’s not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART’s main purpose is to transmit and receive serial data. Introduction to UART  Communication In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART.

UARTs transmit data asynchronously, which means there is no clock signal to synchronize the output of bits from the transmitting UART to the sampling of bits by the receiving UART. Instead of a clock signal, the transmitting UART adds start and stop bits to the data packet being transferred. These bits define the beginning and end of the data packet so the receiving UART knows when to start reading the bits. When the receiving UART detects a start bit, it starts to read the incoming bits at a specific frequency known as the baud rate. Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps).

Both UARTs must operate at about the same baud rate. The baud rate between the transmitting and receiving UARTs can only differ by about 3% before the timing of bits gets too far off. Both UARTs must also must be configured to transmit and receive the same data packet structure.

**ESP32 Pinout for serial**



**ESP32 Hardware Serial2 Arduino Example Code**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | /\*  \* There are three serial ports on the ESP known as U0UXD, U1UXD and U2UXD.  \*  \* U0UXD is used to communicate with the ESP32 for programming and during reset/boot.  \* U1UXD is unused and can be used for your projects. Some boards use this port for SPI Flash access though  \* U2UXD is unused and can be used for your projects.  \*  \*/    #define RXD2 16  #define TXD2 17    void setup() {    // Note the format for setting a serial port is as follows: Serial2.begin(baud-rate, protocol, RX pin, TX pin);    Serial.begin(115200);    //Serial1.begin(9600, SERIAL\_8N1, RXD2, TXD2);    Serial2.begin(9600, SERIAL\_8N1, RXD2, TXD2);    Serial.println("Serial Txd is on pin: "+String(TX));    Serial.println("Serial Rxd is on pin: "+String(RX));  }    void loop() { //Choose Serial1 or Serial2 as required    while (Serial2.available()) {      Serial.print(char(Serial2.read()));    }  } |

This program reads data from serial2 and sends to serial0 i.e. programming serial. Multiple serial is useful when using GPS and GSM systems together.